

Wenqin ZHOU (✉), David J. JEFFREY

Applied Mathematics Department, University of Western Ontario, London, Ontario, Canada

©

P is a permutation matrix, L and U are triangular as shown, and the pivots p_i that arise are in \mathbb{R} . The pivot p_n is also the determinant of the matrix $[a_{i,j}]_{i,j \leq n}$.

Proof For a full-rank matrix, there always exists a permutation matrix such that the diagonal pivots are

non-zero during Gaussian elimination. Let P be such a permutation matrix for the full-rank matrix A (We will give details in the algorithm for finding this permutation matrix). Classical Gaussian elimination shows that PA admits the following factorization.

$$\begin{aligned}
 PA &= \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} & \cdots & a_{1,m} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,n} & \cdots & a_{2,m} \\ a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{3,n} & \cdots & a_{3,m} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ a_{n,1} & a_{n,2} & a_{n,3} & \cdots & a_{n,n} & \cdots & a_{n,m} \end{pmatrix} \\
 &= \begin{pmatrix} 1 & & & & & & \\ \frac{a_{2,1}}{a_{1,1}} & 1 & & & a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} & \cdots & a_{1,m} \\ \frac{a_{3,1}}{a_{1,1}} & \frac{a_{3,2}^{(1)}}{a_{2,2}^{(1)}} & 1 & & & a_{2,2}^{(1)} & a_{2,3}^{(1)} & \cdots & a_{2,n}^{(1)} & \cdots & a_{2,m}^{(1)} \\ \vdots & \vdots & \vdots & \ddots & & & a_{3,3}^{(2)} & \cdots & a_{3,n}^{(2)} & \cdots & a_{3,m}^{(2)} \\ \frac{a_{n,1}}{a_{1,1}} & \frac{a_{n,2}^{(1)}}{a_{2,2}^{(1)}} & \frac{a_{n,3}^{(2)}}{a_{3,3}^{(2)}} & \cdots & 1 & & & & a_{n,n}^{(n-1)} & \cdots & a_{n,m}^{(n-1)} \end{pmatrix} \\
 &= \begin{pmatrix} 1 & & & & & & & & & & \\ \frac{a_{2,1}}{a_{1,1}} & 1 & & & a_{1,1} & & & & & & \\ \frac{a_{3,1}}{a_{1,1}} & \frac{a_{3,2}^{(1)}}{a_{2,2}^{(1)}} & 1 & & & a_{2,2}^{(1)} & & & & & \\ \vdots & \vdots & \vdots & \ddots & & & a_{3,3}^{(2)} & & & & \\ \frac{a_{n,1}}{a_{1,1}} & \frac{a_{n,2}^{(1)}}{a_{2,2}^{(1)}} & \frac{a_{n,3}^{(2)}}{a_{3,3}^{(2)}} & \cdots & 1 & & & & & & a_{n,n}^{(n-1)} \end{pmatrix} \\
 &= \begin{pmatrix} 1 & \frac{a_{1,2}}{a_{1,1}} & \frac{a_{1,3}}{a_{1,1}} & \cdots & \frac{a_{1,n}}{a_{1,1}} & \cdots & \frac{a_{1,m}}{a_{1,1}} \\ & 1 & \frac{a_{2,3}^{(1)}}{a_{2,2}^{(1)}} & \cdots & \frac{a_{2,n}^{(1)}}{a_{2,2}^{(1)}} & \cdots & \frac{a_{2,m}^{(1)}}{a_{2,2}^{(1)}} \\ & & 1 & \cdots & \frac{a_{3,n}^{(2)}}{a_{3,3}^{(2)}} & \cdots & \frac{a_{3,m}^{(2)}}{a_{3,3}^{(2)}} \\ & & & \ddots & \vdots & \cdots & \vdots \\ & & & & 1 & \cdots & \frac{a_{n,m}^{(n-1)}}{a_{n,n}^{(n-1)}} \end{pmatrix}
 \end{aligned}$$

$$i=1, \quad a_{i,j}^{(0)} = a_{i,j};$$

$$2 \leq i \leq j \leq m, \quad a_{i,j}^{(i-1)} = \frac{a_{i-1,i-1}^{(i-2)} \quad a_{i-1,j}^{(i-2)}}{\frac{a_{i-1,i-1}^{(i-2)}}{a_{i-1,i-1}^{(i-2)}} \quad \frac{a_{i-1,i-1}^{(i-2)}}{a_{i-1,i-1}^{(i-2)}}};$$

$$n \geq i \geq j \geq 2, \quad a_{i,j}^{(j-1)} = \frac{a_{i,j-1}^{(j-2)} \quad a_{j-1,j}^{(j-2)}}{\frac{a_{j-1,j-1}^{(j-2)}}{a_{j-1,j-1}^{(j-2)}} \quad \frac{a_{j-1,j-1}^{(j-2)}}{a_{j-1,j-1}^{(j-2)}}}.$$

Let us define $A_{i,j}^{(k)}$ by $A_{i,j}^{(k)} =$

$a_{1,1}$	$a_{1,2}$	\cdots	$a_{1,k}$	$a_{1,j}$
$a_{2,1}$	$a_{2,2}$	\cdots	$a_{2,k}$	$a_{2,j}$
\cdots	\cdots	\cdots	\cdots	\cdots
$a_{k,1}$	$a_{k,2}$	\cdots	$a_{k,k}$	$a_{k,j}$
$a_{i,1}$	$a_{i,2}$	\cdots	$a_{i,k}$	$a_{i,j}$

Hence, these are in \mathbb{I} .

j

j


```

    U[i,k] := 0;
end do;
oldpivot := U[k,k];
end do;
L[n,n] := 1/oldpivot;

```

□

The main difference between Algorithm 1 and Algorithm 2 is that Algorithm 2 uses non-exact divisions when computing the L matrix. The reason we give these two algorithms is that we want to show the advantage of a fraction free output format.

Theorem 3 Let A be a $n \times m$ matrix of full rank with entries in a domain \mathbb{I} and $n \leq m$. On input A , Algorithm 1 outputs four matrices P, L, D, U with entries in \mathbb{I} such that $PA = LD^{-1}U$, P is a $n \times n$ permutation matrix, L is a $n \times n$ lower triangular matrix, D is a $n \times n$ diagonal matrix, U is a $n \times m$ upper triangular matrix. Furthermore, all divisions are exact.

Proof In Algorithm 1, each pass through the main loop starts by finding a non-zero pivot, and reorders the row accordingly. For the sake of proof, we can suppose that the rows have been permuted from the start, so that no permutation is necessary.

Then we prove by induction that at the end of step k , for $k = 1, \dots, n-1$, we have

- $QD[1] = A_{1,1}^{(0)}$ and $D[i] = A_{i-1,i-1}^{(i-2)} A_{i,i}^{(i-1)}$ for $i = 1, \dots, k$,
- $L[i,j] = A_{i,j}^{(j-1)}$ for $j = 1, \dots, k$ and $i = 1, \dots, n$,
- $U[i,j] = A_{i,j}^{(j-1)}$ for $i = 1, \dots, k$ and $j = i, \dots, m$,
- $U[i,j] = A_{i,j}^{(k)}$ for $i = k+1, \dots, n$ and $j = k+1, \dots, m$,
- all other entries are 0.

These equalities are easily checked for $k = 1$. Suppose that this holds at step k , and let us prove it at step $k+1$. Then,

- for $i = k+1, \dots, n$, $L[i,k+1]$ gets the value $U[i,k+1]$
- $= A_{i,k+1}^{(k)}$,
- $D[k+1]$ gets the value $A_{k,k}^{(k-1)} A_{k+1,k+1}^{(k)}$,
- for $i,j = k+2, \dots, m$, $U[i,j]$ gets the value

$$\underline{A_{k,k}^{(k-1)} A_{i,j}^{(k-1)} - AA}$$

and

$$L_{i,k} = A_{i,k}^{(k-1)}, D_{k,k} = A_{k-1,k-1}^{(k-2)} A_{k,k}^{(k-1)}.$$

Lemma 2 If every entry $a_{i,j}$ of the matrix $A = [a_{i,j}]_{n \times n}$ is a univariate polynomial over a field \mathbb{K} with degree less than d , we have $\deg A_{i,j}^{(k)} \leq kd$.

If every entry $a_{i,j}$ of the matrix $A = [a_{i,j}]_{n \times n}$ is in \mathbb{K} and has length bounded by ℓ , we have $\lambda A_{i,j}^{(k)} \leq k(\ell + \log k)$.

If every entry of the matrix $A = [a_{i,j}]_{n \times n}$ is a univariate polynomial over $\mathbb{K}[x]$ with degree less than d and coefficient's length bounded by ℓ , we have $\deg A_{i,j}^{(k)} \leq kd$ and $\lambda A_{i,j}^{(k)} \leq k(\ell + \log k + d \log 2)$.

Proof If $a_{i,j} \in \mathbb{K}$ has degree less than d , from Lemma 1, we have $\deg A_{i,j}^{(k)} \leq kd$. If $a_{i,j} \in \mathbb{K}$ has length bounded by ℓ , from Eq. 3 and Lemma 1 with $m = 0$, we have $\lambda A_{i,j}^{(k)} \leq k(\ell + \log k)$. If $a_{i,j} \in \mathbb{K}[x]$ has degree less than d and coefficient's length bounded by ℓ , from Eq. 3 and Lemma 1 with $m = 1$, we have $\deg A_{i,j}^{(k)} \leq kd$ and $\lambda A_{i,j}^{(k)} \leq k(\ell + \log k + d \log 2)$. \square

In the following part of this section, we want to demonstrate that the difference between fraction free LU factoring and our completely fraction free LU factoring is the divisions used in computing their lower triangular matrices L . We discuss here only three cases. In case 1, we will analyze the cost of two algorithms with $A \in \mathbb{K}[x]$, where \mathbb{K} is a field, i.e., we only consider the growth of degree during the factoring. In case 2, we will analyze the cost of two algorithms with $A \in \mathbb{K}$, i.e. we only consider the growth of length during the factoring. In case 3, we will analyze the cost of both algorithms with $A \in \mathbb{K}[x]$. For more cases, such as $A \in \mathbb{K}[x_1, \dots, x_m]$, the basic idea will be the same as these three basic cases.

Theorem 4 For a matrix $A = [a_{i,j}]_{n \times n}$ with entries in $\mathbb{K}[x]$, if every $a_{i,j}$ has degree less than d , the time complexity of completely fraction free LU factoring for A is bounded by $O(n^3 M(nd))$ operations in \mathbb{K} .

For a matrix $A = [a_{i,j}]_{n \times n}$ with entries in \mathbb{K} , if every $a_{i,j}$ has length bounded by ℓ , the time complexity of completely fraction free LU factoring for A is bounded by $O(n^3 M(n \log n + n\ell))$ word operations.

For a matrix $A = [a_{i,j}]_{n \times n}$ with univariate polynomial entries in $\mathbb{K}[x]$, if every $a_{i,j}$ has degree less than d and has length bounded by ℓ , the time complexity of completely fraction free LU factoring for A is bounded by $O(n^3 M(n^2 d \ell + nd^2))$ word operations.

Proof Let case 1 be the case $a_{i,j} \in \mathbb{K}[x]$ with $d = \max_{i,j} \deg(a_{i,j}) + 1$, case 2 be the case $a_{i,j} \in \mathbb{K}$ with $\ell = \max_{i,j} \lambda a_{i,j}$ and case 3 be the case $a_{i,j} \in \mathbb{K}[x]$ with $d = \max_{i,j} \deg(a_{i,j}) + 1$ and $\ell = \max_{i,j} \lambda a_{i,j}$. From Lemma 2, at each step k , $\deg A_{i,j}^{(k)} \leq kd$ in case 1 and $\lambda A_{i,j}^{(k)} \leq k(\ell + \log k)$ in case 2, and λ

$\log \log (nd(\ell + d)) + ndM(n(\ell + \log n + d) \log (nd(\ell + d)))$
 $\log (n(\ell + d))$.

Proof As above, case 1 is $a_{i,j} \in \mathbb{K}[x]$ with $d = \max_{i,j} \deg(a_{i,j}) + 1$, case 2 is $a_{i,j} \in \mathbb{K}$ with $\ell = \max_{i,j} |a_{i,j}|$ and case 3 is $a_{i,j} \in \mathbb{K}[x]$ with $d = \max_{i,j} \deg(a_{i,j}) + 1$ and $\ell = \max_{i,j} |a_{i,j}|$.
 From Lemma 2, at each step k^{th} , $\deg A_{i,j}^{(k)} \leq kd$ in case 1,
 $\lambda_{i+fk d}^{(k)} \leq \dots \leq \lambda_i^{(k)}$

of the time of completely fraction free LU factoring with the logarithm of the size of matrix (Fig. 2). If we use the Maple command $\text{Fit}(a+b \cdot t, x, y, t)$ to fit it, we find a slope equal to 4.335. This tells us that the relation between the time used by completely fraction free LU factoring and the size of the matrix is $t = O(n^{4.335})$. In the view

In this section, we give applications of our completely fraction free LU factoring. Our first application is to solve a symbolic linear system of equations in a domain. We will introduce fraction-free forward and backward substitutions from Ref. [1]. Our second application is to get a new completely fraction free QR factoring, using the relation between LU factoring and QR factoring given in Ref. [22].

5.1 Fraction free forward and backward substitutions

In order to solve a linear system of equations in one domain, we need not only fraction free LU factoring of the coefficient matrix but also fraction free forward substitution (FFFS) and fraction free backward substitution (FFBS) algorithms.

Let A be a $n \times n$ matrix, and let P, L, D, U , be as in Theorem 3 with $m = n$.

Definition 3 Given a vector b in \mathbb{I} , fraction free forward substitution consists in finding a vector Y , such that $LD^{-1}Y = Pb$ holds.

Theorem 6 The vector Y from Definition 3 has entries in \mathbb{I} .

Proof

$$Y_i = \frac{D_{i,i}}{L_{i,i}} \left[b_{P_i} - \sum_{k=1}^{i-1} L_{i,k} Y_k \right],$$

where $b_{P_i} = \sum_{j=1}^n$

Because $\Theta^T \Theta$ is symmetric and both U and $(DL^{-1})^T$ are upper triangular matrices, $\Theta^T \Theta$ must be a diagonal matrix. So the columns of Θ are left orthogonal and in \mathbb{F}^n based on Theorem 6.

Based on Eq. 7, we have $A^T = LD^{-1}\Theta^T$, i.e., $A = \Theta(LD^{-1})^T = \Theta(D^T)^{-1}L^T = \Theta D^{-1}L^T$. Set $R = L^T$, then R is a fraction free upper triangular matrix. \square

$$L = \begin{pmatrix} 5 & & & & & & \\ 7 & 21 & & & & & \\ & 7 & -19 & 1 & & & \\ 5 & 7 & 7 & 0 & 2 & 0 & 1 \\ 21 & -19 & -10 & 1 & 0 & -2 & \\ & & 310 & -17 & -34 & 21 & 68 \end{pmatrix}, \quad D = \begin{pmatrix} 5 & & & & & & \\ & 105 & & & & & \\ & & 21 & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \end{pmatrix},$$

$$U = \begin{pmatrix} 5 & & & & & & \\ 7 & 21 & & & & & \\ & 7 & -19 & 1 & & & \\ 5 & 7 & 7 & 0 & 2 & 0 & 1 \\ 21 & -19 & -10 & 1 & 0 & -2 & \\ & & 310 & -17 & -34 & 21 & 68 \end{pmatrix}.$$

We can verify that the square of the second row on the right side of matrix U is not equal to the diagonal of the left side of the matrix U . It means the observation of Pursell and Trimble is not valid in this example.

The fraction free QR factoring of matrix C is as follows:

$\ominus D^-$
